AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Previously Presented) A method of accessing state from a configurable processor, the method comprising:

transmitting, using a debugger, a state-accessing instruction stream to an interpreting agent, the interpreting agent being capable of interpreting that stream; and

causing, using the state-accessing instruction stream, the interpreting agent to return the state of the processor to the debugger.

2. (Original) A method as in claim 1 where the interpreting agent is a monitor program.

3. (Original) A method as in claim 1 where the interpreting agent is an instruction insertion server.

4. (Original) A method as in claim 1 where the interpreting agent is an architectural simulator.

5. (Previously Presented) A method as in claim 1, further comprising:

reading, using the debugger, information describing the configurable processor's state architecture; and

generating, using the debugger, the instruction stream based on the information.

6. (Original) A method as in claim 5 wherein the interpreting agent is a monitor program.

7. (Original) A method as in claim 5 wherein the interpreting agent is an instruction insertion server.

2

Newlin et al.
Appln. No. 09/680,126
Amendment – March 3, 2004

TEN-007
Atty. Dkt. No. 83818-269851

8. (Original) A method as in claim 5 wherein the interpreting agent is an architectural simulator.

9. **(Currently Amended)** A computer-readable storage medium storing therein a software program comprising:

    software for **automatically** generating a **hardware description of a** configurable processor from a user description of that processor; and

    a debugger library for **automatically** generating information necessary to describe save and restore instructions for state of the configurable processor based on the user description.

10. (Previously Presented) A computer-readable storage medium storing therein a debugger library for:

    reading a description of save and restore state information of a configurable processor; and

    generating saving and restoring state instruction streams based on the description.

11. (Previously Presented) A medium as in claim 10 wherein the debugger library further comprises functionality for:

    identifying interdependencies in state; and

    generating a complete and correct save and restore sequence based on the interdependencies.

12. **(Currently Amended)** An instruction-insertion server comprising:

    means for retrieving system topology information of a chip containing multiple cores from a computer-readable file; ~~and~~

    means for determining where elements are in a system described by the file**; and**

    means responsive to the determining means for directing a state-accessing instruction stream to an appropriate one of the multiple cores.

3

Newlin et al.
Appln. No. 09/680,126
Amendment – March 3, 2004

TEN-007
Atty. Dkt. No. 83818-269851

13. (Previously Presented) A system for accessing state from a configurable processor, the system comprising:

a debugger which transmits a state-accessing instruction stream;

an interpreting agent which

receives the instruction stream,

interprets the instruction stream to access state of the configurable processor, and

returns the accessed state of the configurable processor to the debugger.

14. (Original) A system as in claim 13 where the interpreting agent is a monitor program.

15. (Original) A system as in claim 13 where the interpreting agent is an instruction insertion server.

16. (Original) A system as in claim 13 where the interpreting agent is an architectural simulator.

17. (Previously Presented) A system as in claim 13, wherein the debugger is adapted to:

read information describing the configurable processor's state architecture; and

generate the instruction stream based on the information.

18. (Original) A system as in claim 17 wherein the interpreting agent is a monitor program.

19. (Original) A system as in claim 17 wherein the interpreting agent is an instruction insertion server.

20. (Original) A system as in claim 17 wherein the interpreting agent is an architectural simulator.

4